Analysis of GBS Phase I Software Capabilities And Its Potential Reusability for GBS Phase II

July 10, 1997

CDRL SEQUENCE NO. A003

Sponsored by:

Defense Advanced Research Projects Agency
Information Systems Office
DARPA Order No. E456
Issued by DARPA/CMO under Contract MDA972-96-C-0025

19970721 111

Prepared by:

Welkin Associates, Ltd.
Suite 210
4801 Stonecroft Blvd
Chantilly, Virginia 20151
(703) 691-4616

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10 July 1997	Final Report

4. TITLE AND SUBTITLE
Analysis of GBS Phase I Software Capabilities And Its Potential Reusability for GBS Phase II

5. FUNDING NUMBERS
MDA972-96-C-0025

6. AUTHOR(S)
M. Snellings, M. Giboney, K Dezell, J. Nyikos

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Welkin Associates Ltd.
4801 Stonecroft Blvd. Suite 210
Chantilly, VA 20151

8. PERFORMING ORGANIZATION REPORT NUMBER
0025-CDRL-003

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Information Systems Agency (DISA)
DITCO-NCR
701 South Courthouse Road
Arlington, VA 22204-2199

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
None

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This document analyzes GBS Phase I software capabilities and its potential re-use in the development of the GBS Phase II System. It is intended to assist the GBS Joint Program Office and the development contractor in their effort to design and build the Phase II system.This analysis evaluates the GBS version 2.1 software which was deployed to the JBS sites in during the spring of 1997.

DTIC QUALITY INSPECTED 4

14. SUBJECT TERMS	15. NUMBER OF PAGES
GBS Phase I Software Reusability	20
	16. PRICE CODE: N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

TABLE OF CONTENTS

1. Executive Summary

As the Global Broadcast Service (GBS) enters its second phase of development, a determination must be made as to the reusability of the Phase I software. The Phase I software was developed over the last several years as a prototype and is currently operational as the Joint Broadcast Service (JBS) supporting the Bosnia Command and Control Augmentation (BC2A).

The potential reusability of the GBS Phase I software will be determined in large part by the platform and operating system selected for the Phase II receive suites. The two most likely candidates are the continued use of the Sun Sparc workstations with the Solaris operating system or the use a PC running Windows NT.

If the existing Sparc platform is used for Phase II, much of the Phase I software *can* be reused, although this is not recommended. The Phase I software was developed primarily as a prototype and is not of the quality that one expects from a formal development effort. It is also felt that much of the existing custom software can and should be replace by COTS software which would be better able to provided the necessary system capabilities, while reducing the risks inherent in the use of custom software.

If the PC/Windows NT platform is selected for Phase II, reusing the Phase I software would require porting or redeveloping the majority of the Phase I software for use under Windows NT. Rather than porting software which supports a Phase I architecture, the complete system should be re-architected, so that a design which is right for the PC and Phase II requirements is implemented.

In either case, the queue manager software is viewed as one piece of software which could be reused given a Sun platform or ported given a Windows NT platform. This software was integrated into the GBS system based on its success within other programs and has proven to be work very well.

Given the choice of the above mentioned receive suite platforms, I recommend that the Windows NT platform be selected based primarily on its ease of use and administration for non technical users of the system.

Each custom application in the Phase I system should reviewed by the Phase II software development contractor so that a clear understanding of the Phase I capabilities is obtained. The Phase II system should, at a minimum, satisfy the capabilities implement in Phase I plus those specifically identified as Phase II requirements. Since Phase II strongly emphasizes the use of COTS software, effort should be expended to replace much of the Phase I custom software with COTS.

In short, most of the Phase I software should not be reused in it's current state, instead its architecture should be reviewed and considered in developing the Phase II software architecture.

2. Scope

This document analyzes the GBS Phase I software capabilities and its potential reuse in the development of the GBS Phase II system. It is intended to assist the GBS Joint Program Office (JPO) and the Phase II software development contractor in their effort to design and build the Phase II system.

This analysis evaluates the GBS version 2.1 software which was deployed to the JBS sites during the spring of 1997. The reusability of this software is evaluated based on the continuing requirements for the Phase I capabilities and the documented Phase II requirements.

This document explains the capabilities of the Phase I software, the need for these capabilities, and how the software can and should be reused.

2.1 Overview

The first part of this document focuses on reviewing the requirements for the Phase II. These requirements determine which Phase I capabilities will be useful in the future system. Since the requirements provide flexibility to the Phase II software development contractor, several architectures are discussed. The reuse of the existing software depends largely on the architecture chosen.

The analysis of the Phase I software is divided into functional areas. These functional areas are further divided into the broadcast of file-based products, data streams, and multimedia data. Other Phase I capabilities are also discussed.

2.2 System Overview

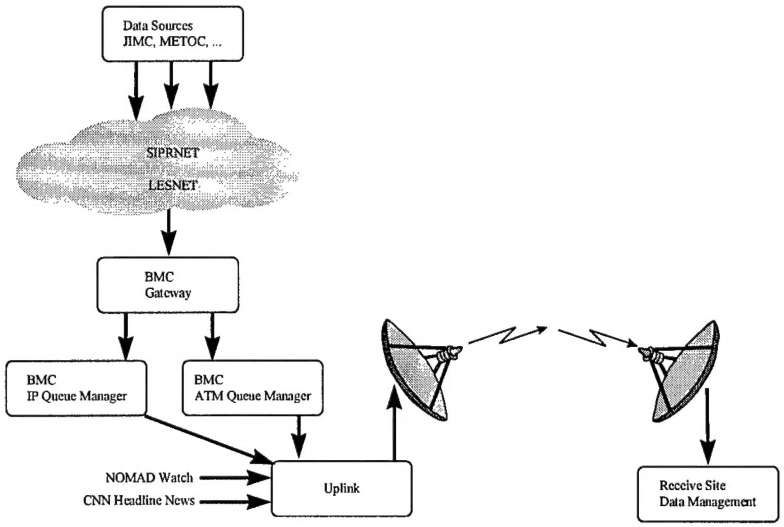
GBS seeks to take advantage of commercial communications technology to provide intelligence data to the intelligence consumer in a timely fashion. It is important to note that GBS itself is NOT an intelligence system but the mechanism to provide intelligence information to other users, either human or intelligence exploitation systems. Of particular importance is the fact that GBS can provide LARGE amounts of audio, video, and data, up to 30 Mbps, to locations which previously could not receive such information due to physical or bandwidth constraints. This large bandwidth can be achieved at a fraction of the cost of custom built communications system with a minimal set of hardware. High bandwidth with minimal hardware constraints opens the door for providing intelligence information to the warfighter in the lower echelons. It also means that intelligence, such as video and imagery, which may have been previously unavailable to the warfighter may now be provided.

GBS has been designed to satisfy the increasing demands for greater bandwidth, real-time video, and high speed data transmission. It is capable of providing diverse types of information to a large audience of tactically deployed military users within a designated geographic area such as Bosnia and the surrounding countries. Utilizing the rapidly expanding field of commercial Direct Broadcast Satellite (DBS), GBS is able to disseminate images, large data files, and other high bandwidth information concurrently to a large number of widely separated user locations.

The development of GBS is being performed in a phased approach. Phase I consists of the capabilities currently fielded in support of Operation Joint Guard and the CONUS based GBS Testbed. This prototype development effort has proved the GBS concept. Phase II will formalize this effort.

2.3 Software Overview

There are three basic architectural components of the GBS system. The first is the GBS data source which is a provider of the data files to be included in the broadcast. The second is the broadcast management center (BMC) which takes data files from the source and includes them in the broadcast uplink. The third component of GBS is the receive site where data from the broadcast is accepted and processed The following figure shows the basic GBS architecture for data file processing.



The GBS software supports both IP and ATM protocols. These protocols define the physical connectivity between the BMC workstation and satellite transmitter, and the satellite receiver and Receive workstation. The data broadcast over each channel (e.g., ATM US Secret data) uses a single protocol; the channel includes file, stream, and status data combined in single data channel. The software used to support the IP and ATM transport protocols is identical, with several run time configuration file differences.

2.4 Government documents

The following documents have been used as reference material during the software evaluation. They should be consulted for further details about concepts described in this document. Of particular interest is the GBS Software Architecture Document which provides extensive details about the software capabilities and implementation.

Item	Document Identification	Title
1	30 January 1997	GBS Software Architecture Document, Version 2.1
2	8 May 1997	Interface Requirements Specification (IRS) for the Global Broadcast Service
3	8 May 1997	GBS Phase II System Requirements Document (SRD)
4	7 May 1997	GBS Systems Contract, Section M
5	10 March 1997	Global Broadcast Service Version 2.1 Release Notes
6	28 March 1997	GBS Phase I Lessons Learned
7	1 April 1997	Global Broadcast Service Receive User's Guide

2.5 Existing Software

The GBS version 2.1 software baseline contains approximately 100,000 source lines of code. The software is written in a variety of different languages by many different development groups. As a result, there is limited software reuse within the entire GBS system. Also, a significant portion of the software is not

written using object oriented techniques. The following table describes the distribution of software across different programming languages.

Language	Lines of code
C	45%
C++	20%
Generated C from UIM/X	10%
Script (Bourne, C, Perl)	20%
HTML	5%

Although the baseline contains approximately 100,000 lines of code, some of the code implements duplicate functionality which has not been removed from the baseline. This duplication of functionality provides a meaningful way to evaluate different system approaches and capabilities. The two primary examples are the two different queue managers and two different implementations of file transport software. Selecting one implementation and removing the duplicate functionality would remove approximately 10,000 lines of code from the baseline.

This section describes the Commercial Off The Shelf (COTS) software, Government Off The Shelf (GOTS) software, and Other Off The Shelf (OOTS) software used to support GBS software development and deployment. The following lists describe the Version 2.1 GBS Software Release requirements.

Required COTS software and licenses to support development
One software license is required for each CPU upon which this software package is used for software application development. There is no runtime license requirement for applications developed with these COTS software packages.

- OSF/Motif, version 1.2.4 (Bluestone Consulting Inc.)
  Motif window manager and development libraries
- IXI Motif, version 1.4 (IXI Corporation)
  Motif window manager and development libraries. This is required for some of the GUI programs since the Solaris and Bluestone X Window servers have memory leak problems. This could replace the OSF/Motif software from Bluestone Consulting.
- UIM/X, version 2.9 (Bluestone Consulting Inc.)
  X Window Graphical User Interface (GUI) Builder
- XRT/Gear for Motif, version 1.0.1 (KL Group)
  X Windows widget set including tool bar, tool tip, folder widgets
- XRT/Table for Motif, version 2.2.0 (KL Group)
  X Window widget set including spreadsheet type table widget
- Visual Work Shop, version 2.1 (SunSoft)
  C++ programming environment which includes compiler, debugger, and memory management.

Required OOTS software to support development
This software is required for software development, but is not distributed with the GBS software.

- gcc, version 2.7.2 (GNU)

Each of the following software packages are maintained in the GBS baseline. A copy of the software is placed under the CM structure, but the software is not maintained under a configuration management system (e.g., SCCS).

Required COTS software to support deployment

- Netscape, version 2.02 (Netscape Communications Corp.)
  This program is distributed to support analysis of data transmitted via the GBS broadcast. The BC2A program has purchased a site license for 100 systems.

- UNIFLIX Audio and Video transmission software, version 3.3.1 (Paradise Software, Inc.)
  Paradise Software has made many enhancements specifically to support GBS; the version number may be incorrect depending whether all changes have been rolled into an official product release.
  The BC2A licensing agreement includes a site license for the client software (receive workstations), the server software (BMC workstations) are individually licensed
- Fore ATM device driver, version 4.0.0 (Fore Systems), required only for ATM systems
  This software is distributed with the Fore ATM sbus cards.
- Parallax video device driver (Parallax Graphics, Inc.), required only for ATM systems
  This software is distributed with the Parallax video sbus cards.
- Sun Net Manager, (Sun Soft), required only for ATM systems
  BC2A has a single site license which covers the entire GBS network through the receive workstations.
- MIBs, (Yurie Systems, Inc.), required only for ATM systems

Required GOTS software to support deployment

- Client Server Environment (CSE), version
  Enhances Solaris system security.
- Oilstock, version 4.3
  This program is distributed to support analysis of data transmitted via the GBS broadcast.
- Matrix, version 4.0.2
  This program is distributed to support analysis of data transmitted via the GBS broadcast.

Required OOTS software to support deployment

- perl, version 5.002 (Larry Wall / Internet)
  The baseline contains several scripts written in Perl, particularly software supporting the web interface.
- httpd, version 1.5.2 (NCSA)
  This is the web server.
- xv, version 3.10 (John Bradley / Internet)
  This program is distributed to support analysis of data transmitted via the GBS broadcast.
- Acrobat Reader, version 3.0 (Adobe)
  This program is distributed to support analysis of data transmitted via the GBS broadcast.

3. GBS Phase II Software Requirements

This section provides an overview of the GBS Phase II software requirements extracted from various reference documents which can impact the reuse of GBS Phase I software. The GBS Phase II documentation does not contain detailed software requirements, but instead identifies generic capabilities which are expected of the Phase II software.

For the purpose of this document, it is assumed that the existing capabilities of the Phase I software are requirements for the Phase II system. Only those requirements that are partially or fully satisfied by the Phase I software, or which relate to existing capabilities of the Phase I software are detailed here. This section is not meant to summarize all GBS Phase II software requirements, but only those which may affect the reuse of the Phase I software.

The requirements which **significantly** impact the reuse of the existing Phase I software are:

- Maximize the integration of Commercial Off The Shelf (COTS) software
- Provide seamless network connectivity giving significant bandwidth improvements

These requirements and their impacts are discussed in the following sections.

3.1 COTS Software

The Phase II requirements call for maximizing the use of both COTS and Government Off The Shelf (GOTS) software. In many cases, shareware products costing only $10-$20 and used by millions of users can be integrated into the system to provide significant functionality.

The use of COTS/GOTS software is beneficial for several reasons. The government is able to leverage development costs across a larger group of consumers which results in software which costs less than custom software while simultaneously providing more features. COTS/GOTS software is typically more mature software due to the larger pool of users identifying problems. Also commercial software vendors have a rapid problem isolation/resolution cycle, often forced by upon them by competition. Another benefit is the ability to take advantage of new capabilities as newer versions of the COTS/GOTS products become available. The cost of upgrading and integrating a revised product is minimal compared to the cost of writing custom software to implement new capabilities.

The Phase I system uses Uniflix COTS software to provide the audio and video data transmission for GBS. This provides the classified audio feed which accompanies the unclassified Predator video as well as classified audio/video telecasts such as DOD briefings. Additionally, all of the data viewing programs which are distributed with the GBS software are either COTS or GOTS products. These programs include Xview, Netscape, Acrobat Reader, Matrix, and Oilstock

One of the most important goals for the Phase II software contractor is to replace as much of the existing custom software and to minimize the yet to be written software by integrating as many COTS/GOTS products as is technically justified.

3.2 Network Integration

One of the goals of GBS Phase II is to perform better integration of the GBS capability into the user environment. GBS Phase II will be integrated into the Defense Information System Network (DISN). The optimal approach is to remove (from the transport layer and hopefully the user interface) the need to specify meta data (which is currently done by wrapping each file transmitted) for all file data transmitted via the GBS system. The use of this non-transport meta-data needs be reconsidered when designing the Information Dissemination Management (IDM) portion of GBS. The existing GBS system does not include IDM capabilities.

Most users have full network connectivity into SIPRNET. GBS provides the capability to significantly increase users bandwidth to SIPRNET in one direction. That is, SIPRNET traffic destined for GBS users can be forward via the high bandwidth GBS link instead of using slower terrestrial communication lines. This capability can be accomplished by using the split IP concept where data packets traveling in one direction take a completely different path than the packets traveling the reverse direction (e.g., one direction utilizes the GBS broadcast while the other direction traverses the SIPRNET backbone). The GBS broadcast will create a high speed/bandwidth bridge for DISN data destined for remote users. With the GBS integration into DISN, GBS users with full network connectivity will have support for applications using standard protocols such as File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Hyper-Text Transfer Protocol (HTTP), Domain Name Server (DNS), and others.

The split IP concept is already available commercially to surf the web. The slow network connectivity is provided through a standard modem connection to an internet provider while the high speed / high bandwidth connection is provided by a digital satellite system. The implementation of this capability in GBS would probably require the addition of applications to perform routine of IP packets. Additionally, the queue manager software would need to be enhanced since a constant data rate would no longer be available to broadcast files. After GBS handle the forwarding of data packets, the queue manager would need to automatically adjusts its data rate in order to avoid a data overflow on the broadcast.

The Phase II GBS system must provide standard networking capabilities, especially for sites with bi-directional communication capabilities. Although the system can be more robust for fully networked sites, the capability to broadcast data in a simplex fashion to remote sites should not be removed.

4. Phase II Platform and Operating System Selection

Given the requirements for the GBS Phase II software, there are many approaches for implementing the intended system. These approaches range from augmenting the existing software and hardware platform with new capabilities as required to completely redeveloping the system on a PC running Windows NT.

The reusability of the existing software varies based on which approach is chosen. In any case, the analysis contained in this document deals not only with the reuse of the existing code, but also how the software architecture selection impacts the potential reusability.

Several approaches for a Phase II system are detailed in the following sections. These approaches are centered around the use of a Sun workstation or a PC for the hardware platform and the use of Solaris or Windows NT as the operating system. The following architectures are examined:

- Evolve software on Sun UNIX workstation
- Port software to Solaris to the x86 platform
- Port software to Windows NT
- Rewrite software for Windows NT

These are not the only possible solutions, but instead are approaches that seem most likely to be implemented. The following sections discuss the merits and probability of each architecture choice. The selected approach should probably be a combination of the two Windows NT approaches. A thorough COTS evaluation should be performed, and then the remaining required custom software should be ported if possible.

4.1 Sun Receive Workstation

Continuing the deployment of GBS on Sun workstations running the Solaris operating system maximizes the potential for software reuse during Phase II development. However, based on the lessons learned from Phase I, this is not an ideal solution.

As stated in the GBS Phase I Shortcomings: "The Phase I system is hard to use, not intuitive, and manpower intensive. Users are oriented toward commercial personal computer paradigms and expect the GBS system interfaces to react similarly. Phase I operators are often forced to act as system administrators or employ UNIX gurus to support daily maintenance of their receive terminal." We have learned from Phase I that the current software solution is not adequate.

The lessons learned are based on the GBS version 1 software which was originally fielded in spring 1996. Many of the problems identified with this software have been resolved with the deployment of the version 2.1 software. Two major enhancements in the version 2.1 software are that the Graphical User Interfaces (GUIs) now has more of a PC look and feel and the need for frequent maintenance has been resolved. Even with this new GUI, significant changes to the user interface are needed to make the system DII COE compliant.

The general concern of deploying UNIX based systems to sites without a UNIX knowledgeable person continue to exist. Even though, the DII COE Common Desktop Environment will help isolate users from typical UNIX administration chores, a detailed knowledge of UNIX would still be a necessity. This may be enough to prevent the selection of a Sun based platform for Phase II.

4.2 Solaris Operating System on PC Platform

Using an the GBS version 1 software, the Operational Support Office (OSO) demonstrated that the Phase I receive software hosted on a PC platform running the Solaris operating system provides identical capability and similar performance to a Sun workstation based receive system. This approach provides the same reuse potential as a Sun / Solaris platform, however it also provides a significant benefit over the Phase I system: the hardware is less expensive.

Though this approach may be more beneficial than the continued use of the Sun workstation, it still utilizes the same operating system and GBS software identified in the previous approach, and does not solve the usability problems from Phase I. For this reason, this approach is not advised. Hosting GBS software on a Solaris based PC does not make the GBS software any more/less reusability compared to the continued use of the Sun workstation implementation.

4.3 Port Software to Windows NT

Porting the existing software to PC hardware running the Windows NT operation system is an ideal approach for several reasons. It allows the receive workstation to be developed and deployed on a less expensive hardware platform. In addition, it allows the user to operate in an environment in which they are more accustomed. Many of the standard features of the Windows NT operating system and standard programs can be directly accessed.

A significant advantage of selecting the Windows NT operating system is that it provides much better support for tasks which are difficult for GBS users to perform under Solaris, including:

- User Account Maintenance
- Loading/Removing Software
- Deletion and Movement of Files on Disk
- Viewing/Managing of Log Files
- Consistent Application Interfaces
- System Backup/Restore
- Consistent Help Facility

Even though UNIX has support to perform all of these tasks, there is no consistent means used throughout the system or among a group of UNIX systems and they are not centralized, as is the case in Windows NT. This issue will be minimized by the inclusion of CSE or DII COE in the Phase II software architecture. Both packages provide system security and as a by product provide similar system administration functionality across both UNIX and Windows NT platforms.

The Phase I software architecture can cleanly port to a Windows NT platform. The ease with which the actual software can be ported to the Windows NT operating system depends on the nature of the program. Most of the non-GUI applications can be ported without much difficulty. The GUI programs that were developed with using a GUI builder tool could potentially be ported by using the GUI builder tool to translate to the Windows NT APIs. However, this approach would not provide clean software which automatically inherits much of the look and feel of the Windows environment.

This approach provides the fastest way to develop a GBS for a Windows NT platform and would allow a modest amount of software reuse. The GUI portion of the custom applications should be re-written to the Windows APIs. The selection of this approach should be based on the amount of COTS software that can be incorporated into GBS to replace existing functionality.

4.4 Re-Architect and Develop Software for Windows NT

The most work intensive approach is to completely rewrite the GBS software for the Windows NT based system. This means ignoring the Phase I software and designing the system from scratch given the requirements that have been outlined. This approach provides the same benefits of Windows NT over UNIX as discussed above.

Since the Phase I software was developed as a prototype with new capabilities being added as time went on, this approach merits serious discussion. An analysis of the current Phase I system would need to be performed so that the Phase II requirements could be augmented to fully specify the necessary Phase I capabilities.

This approach provides an opportunity to perform a serious analysis of available COTS software which could replace existing functionality. After the necessary packages are selected, it could then be determined what custom software would need to be developed. The COTS evaluation is the most beneficial aspect of this approach.

Portions of the Phase I software, such as the Queue Manager, should be considered when redeveloping the system. This and other portions of the software have demonstrated their ability for reuse, even to other platforms. Since the Queue Manager is such an important piece of the overall software using it as segment in a newly designed system must be considered.

5. Analysis of GBS Phase I Software

This section provides an analysis of the existing GBS Phase I software. It is subdivided into sections analyzing the software functionality, user interface, performance, and quality.

The analysis which is offered here is subjective based on an interpretation of the Phase II requirements and perceived preferences outlined in Section 3. Additionally, the usefulness of the existing software assumes that the Phase I requirements will be satisfied with a software architecture consistent with that of the Phase I software architecture. In order to maximize the reuse of Phase I software, a UNIX based operating system with the X-Windows windowing system would need to be used.

5.1 Functionality

The Phase I software provides for the dissemination of file-based data, system status messages, stream data, and multimedia. These capabilities are considered a subset of the capabilities which are to be include in Phase II software.

5.1.1 File Dissemination

In the Phase I system, every file that is to be disseminated via the GBS broadcast must be "wrapped". Wrapping a file consists of identifying the file to disseminate and specifying the appropriate meta-data. This meta-data currently consists of addressees, classification, priority, type of file, and other optional information. Once this information has been specified, it is packaged with the data file and is transfered to the BMC for broadcast. For a detailed description of the file wrapping process refer to the GBS Receive Users Guide.

The file dissemination architecture is an ideal place to insert COTS software and remove custom code. There are several other ways of disseminating files which are currently used on the Internet which may be more appropriate than individually wrapping files and broadcasting them.

One potential method of file dissemination is the use of Usenet type news servers software which would allow data producers to post files to a news server. The news server residing at the BMC would extract new 'news articles' and broadcast them. After the files are received by the GBS receive workstation, the articles would then get re-posted to the local news server at the GBS receive site.

The use of news servers or any other Internet based file sharing mechanism has not been investigated in great detail. A major advantage in using a news server would be the elimination of a significant amount of custom software that is currently in Phase I for this purpose (close to one half of the custom software in GBS is dedicated to file transport). It would also provide the user with a means of viewing files (news articles) with which they may already be familiar.

The complete integration of GBS into the network will allow the standard protocols, such as ftp and http, to be used while gaining the advantage of the higher bandwidth provided by GBS. This should reduce the number of files that need to be broadcast in the traditional way.

5.1.1.1 File Wrapping

The following section evaluates the different approaches available to wrap files. Currently there are several ways to wrap a file for injection into the GBS broadcast:

- X-Windows application where the user specifies the meta-data via a graphical interface
- Command line based application where the user specified the meta-data on a UNIX command line
- HTTP-based approach which requires the use of a browser such as Netscape
- Automatic wrapping at the BMC

The current means for wrapping files allow users to specify the file to disseminate and the relevant transport information such as classification, priority, and addresses. In addition, other non-transport related information such as keywords, file description, and file permissions may also be specified. The wrapper fields have been expanded in previous releases to allow the user to enter additional meta-data which is then used to perform file filtering at the receive end of the broadcast.

The non-broadcast related information is not necessary for the transport of the file and therefore should be separated from the GBS file and become part of the Information Dissemination Management (IDM) segment. The meta-data is prepended to each file as it is wrapped, this needs to be reengineered so that the meta-data is managed separatly which will eliminate the need to make copies of each file as it is being wrapped and unwrapped.

Many of the wrapper fields currently defined should be removed to simplify the interfaces and to limit the meta-data to transport specific information. Other applications wishing to pass meta-data along with their files should use the user-definable fields which are to be passed along with the file. The X-Windows program and automatic wrapping capability can both be removed since this functionality is provided by the HTTP-based wrap method. Potentially all of the wrapping approaches could be replaced by the a COTS approach such as newsgroup posting. If a COTS solution is not found, then the current means of wrapping files should be provided with a strictly HTTP-based wrapping capability that can be invoked either though a "Wrapping Home Page" or through the execution of a URL from a script, where the URL contains the necessary information for disseminating the file.

It is recommended that the current means for wrapping files be replaced with the use of COTS software such as what is used by the Usenet new servers. The sections below contain the analysis of the existing wrapping applications and the recommendations are based on the continued use of the existing inject capability instead of using some new COTS product for injecting files.

5.1.1.1.1 X-Windows Based Wrapping

This program provides one of two graphical interfaces to support file wrapping. The program provides a window to select which file should be wrapped and another window to specify the meta-data.

The problem with providing a distributed application (whether it be an X-Windows or command line interface) is that each workstation which injects data into the GBS broadcast must have a copy of the wrapping software. As new releases of the software are produced, or as the configuration files are updated (e.g. adding new sites), each copy of the wrapping software must be upgraded. This task is nearly impossible, because not all producers register the fact that they have the software. Even if they do register, is difficult to coordinate this upgrade.

This software could be reused in a Solaris environment, however it would need to be revised to ensure that it is portable to UNIX operating systems other than Solaris. Although the program could be ported to Windows NT, it is not worth the effort since it provides duplicate functionality.

The HTTP-based interface for wrapping files should be provided as the single interface. The URL used to wrap files should be maintained as a managed Application Program Interface (API) of the GBS system. The command line interface should either be converted to a stub program which provides an interface to the HTTP-based URL or can be eliminated entirely by having source sites submit files based on the published wrap URL API.

5.1.1.1.2 Command Line Based Wrapping

The command line based application for wrapping files is used primarily in scripts which provide an automated means of injecting data into the broadcast. It is mainly used by other systems which inject data without human interaction. An example of its use includes the BC2A Information Management System (IMS) which automatically responds to user's requests for data, wrapping the appropriate file and submitting it for broadcast. Additionally, the scripts supporting the HTTP-based wrapping interface calls this program to perform wrapping of selected files.

An alternative to this application to support data source sites is the use of a HTTP-based URL for wrapping files. This URL could be used with scripts and all configuration files would be centrally located at the BMC. This interface is currently being analyzed by the BC2A due to the difficulties which have been experienced with maintaining configuration files and with the integration of GBS wrapping software into other systems such as the Image Product Archive (IPA).

After the Phase II wrapper fields are defined and the approach for storing the meta-data is determined, then the need for this program should be re-evaluated. The only reason that it would need to be maintained is to support the scripts for HTTP-based wrapping. However, if the wrapper is sufficiently simplified, then the functionality could be merged into these scripts since there would not be a need to support different wrap implementations (e.g., scripts, GUI programs, etc.). If the program continues to be necessary, then it will need to be modified to support the new wrapper definition and should also be made platform independent at the same time.

5.1.1.1.3 HTTP-Based File Wrapping

A HTTP-based wrapping capability was introduced into GBS for the Joint Warrior Interoperability Demonstration (JWID) 96 and proved to be a desired capability. This capability provides a user with the ability to wrap data files for GBS dissemination without the requirement for GBS specific software to be resident on the user's workstation. A hyper-text transfer protocol (http) deamon is established on the GBS gateway which allowed users on the attached network, SIPRNET, to enter a specific GBS gateway URL from their Web browser. After entering a userid and password, the resulting page provided an interface for the user to specify a data file and the appropriate meta-data for the file broadcast on GBS.

There are several advantages of this wrapping capability above the rest. First and foremost is that no software, except for a COTS browser, is required to wrap files for GBS dissemination. Another advantage is that wrapping of files becomes platform independent. Where as the X-Windows and command line based wrapping were only available on Sun workstation running Solaris, this wrapping capability is available on any workstation for which a Web browser is available.

Using the X-Windows and command line based wrapping utilities, there was limited security in place to track which users on which workstation have wrapped files. This is because all files that are wrapped on a workstation are transfered to the GBS gateway via ftp using a host specific ftp user id. Using the HTTP-based wrapping, access to wrapping Web page can be controlled on a user-id and/or host basis.

The HTTP-based interface provides the most flexibility wrapping capability of those that currently exist. If the current wrapping/inject architecture is maintained for Phase II, this capability should be maintained and expanded to support growing user requirements.

5.1.1.1.4 Automatic File Wrapping at BMC

Another capability which the Phase I software provides is the ability for the GBS gateway to receive unwrapped files, via ftp, for injection into the broadcast. With this capability, the wrapping criteria is predetermined and stored on the gateway. Data files are transferred to predefined directories using ftp and are automatically wrapped upon arrival and placed into the transmit queue.

This capability allows producers of data to submit files to the BMC without requiring any GBS software on the producer's system. Coordination is done to generate the default meta-data and system access is granted to the producer of the data. The primary purpose of this capability is to minimize the effort required by organizations who are using GBS not for their benefit, but for the benefit of the GBS users.

Each data producer that provides data for GBS broadcast utilizing this method (the METOC is a good example) has a script that causes the file to be transferred to the GBS gateway. The need for automatic file wrapping can be removed by changing the existing script to wrap the file through the HTTP-based URL. This provides the same capability without the need to distribute software to the data producer.

5.1.1.1.5 Transfer of Wrapped Files to BMC

When files are wrapped using the X-Windows or command lined based wrapping, the wrapped files are placed in a directory on the local workstation, where they wait to be transferred to the GBS gateway for injection into the broadcast. The xferit script polls this directory every minute looking for wrapped files which need to be transferred to the BMC. For each file found, an ftp session is created with the GBS gateway, using a userid and password specific to the local gbs user. When the file is successfully transferred, the wrapped file is removed from the local workstation and the next wrapped file is then processed.

The use of ftp to transfer files from the "wrapping" workstation to the GBS gateway is a crude yet reliable way of transferring the files to the BMC. Unfortunately, there are several problems with this approach, the most severe of which is the fact that the existence of these ftp accounts creates a security hole on the GBS system. These accounts can be used by individuals to ftp into the GBS gateway and gain access to files to which they would not otherwise have access.

Several alternate approaches are currently being investigated. These include the use of Usenet like news server so that the wrapped files can be posted to a news server resident on the GBS gateway. Access to the news server would be userid/password controlled in order to maintain control of users having the ability to inject files into the broadcast. Another approach is replacing the ftp submission with a URL based submission. Alternatively, the HTTP-based wrapping can be used exclusively, in which case data files would be transferred to the GBS gateway as part of the http session used for wrapping the file.

Although the need for a method to deliver data files to the BMC will continue to exist, continued using of the current wrapping/inject architecture should demand changes in current approach of using ftp to deliver these files.

5.1.1.2 BMC Queuing of Wrapped Files

As wrapped files are received at the GBS gateway, a determination is made as to when the file should be broadcast, how often, and for how long, if at all, the file should be rebroadcast. When it is determined that a data file should be broadcast, a determination is also made as to which data transport should be used to broadcast the file.

5.1.1.2.1 Timed Broadcast of Data Files

The GBS Phase I software provides the user with the capability, when wrapping files, to optionally specify a date and time that the files should be broadcast. The user can also indicate an interval for rebroadcast of the data file.

The ability to specify the time a file is to be broadcast has been used in the past when disseminating files to a site which is currently down. This capability would allows a user to specify a time in the future for the file to be broadcast (i.e. a time when the site will be up).

The capability for specifying an interval for automatic rebroadcast has also been used when it is known that a site is down but it was not known when it be operating again. By specifying an interval for rebroadcast, it is virtually assured that the file will be received.

These timed broadcast specifications are useful due to the lack of an assured delivery mechanism. The Phase II system should attempt to implement an assured delivery capability. Given that an assured delivery capability exists, the user should not have access to the timed broadcast capability. The users should be able to inject a file with the belief that the system will do whatever possible to deliver the file. This need for this capability may continue to exists, even as the attempted means of assured delivery.

5.1.1.2.2 Determination of Data Channel to Broadcast File

Once it has been determined that it is time to broadcast a file, it must be determined which transport to send the file over (e.g., which queue manager(s) must process the file). This determination is based on the addressees specified during file wrapping and a configuration file which identifies the transport(s) for reaching each destination. This process can be utilized in Phase II so that queues identified in the configuration file for a site are satellite/transponder specific instead of transport specific. For example, instead of having transport values of IP and/or ATM for a destination, the entry may contain UFO8-1 indicating that the queue manager generating data for UFO8 transponder 1 is responsible for delivering a data file.

Since the applications which run on the Gateway workstation contains no user interface and is not tightly coupled with the operating system, the applications performing this determination are highly reusable and should be considered for reuse in Phase II.

5.1.1.3 Broadcasting of Data Files

The GBS Queue Manager is responsible for accepting queued files and broadcasting them using a protocol which is known to both the Queue Manager and the receive workstation. For GBS release 2.1, the MTN Queue Manager was integrated into the system replacing the original OSO Queue Manager. This queue

manager was integrated based on it success with other communications system requiring the broadcast of data files.

The MTN Queue Manager software which has been integrated in to GBS has proven to be very stable software. Efforts are currently underway to port the receive portion of the MTN transport to work on Windows NT, and it believed that porting the Queue Manager portion could be done with similar ease and is recommended as opposed to redeveloping this capability.

5.1.1.4 Receiving Data Files

On the receive end of the broadcast, an MTN process, known as *mar*, receives data packets and reconstructs the data files. Once a data file has been received in its entirety, the *dispose* process interrogates the meta-data in the file wrapper and compares it with a configuration file to determine what action should be taken with the file. The file is either discarded or saved in a directory specified by the configuration file entry.

The concept of receiving files and placing them into directories for other processes or users need to be reworked for the Phase II system. The GBS system is a communications system, which upon receipt of a file should deliver it according to the addressing information associated with the file. It should not be the warehouse for these data files, nor should it rely on platform specific interfaces such at Sun's Network File System (NFS). A multi-platform COTS product, such as the news server previously mentioned, should be used for the dissemination of these files. An optional COTS product could also be utilized for the warehousing of this data for ad-hoc access to it.

5.1.1.5 Receive Suite File Management

The GBS system provides tools for reviewing a list of all files that have been broadcast. This list can be accessed via an X-Windows application known as the GBS File Manager, or via a Web browser. The list shows a subset of the meta-data for each file received in the broadcast, regardless of whether the files were discarded or saved. The file manager tools allow the user to perform queries of this information so that information on specific files can be accessed.

The two receive file managers described in the following sections should both be replaced with a single COTS product which is accessible from a variety of platforms. A news reader, such as that which is embedded in Netscape, can provide similar functionality without the large amount of custom code dedicated to the current file managers.

5.1.1.5.1 X-Windows Receive File Manager

The X-Windows Receive File Manager is one way of viewing information on the data files received via the GBS broadcast. This tool is the primary interface the user has with the GBS receive workstation. As each file is received the list is automatically updated. This tools allows the user to view virtually all meta-data information associated with the data files. In addition to displaying information associated with the files, it allows the user to launch the application for viewing the contents of the data file.

Though this tool does provide the user with access to information on each data file, the integration of a COTS products, such as a Usenet like news server, for managing received data will dictate what tools are used to access received files. For this reason, this application has minimal reuse potential.

5.1.1.5.2 Netscape Receive File Manager

The Netscape Receive File Manager allows the user of the GBS receive suite or a user on a workstation connected to the GBS workstation via a LAN, to access the associated file information via a Web browser. This capability allows user to view the information without requiring physical access to the GBS workstation.

Once the means for storing incoming files is revised, this Netscape Receive File Manager should be replaced with a COTS product providing cross platform support. Such a tool would be the use of the news capabilities build into the Netscape Web browser.

5.1.1.6 File Retransmission

Receive sites that have a network connectivity to the BMC can request retransmission of files previously broadcast. There are two types of retransmission requests, automatic and manual. An automatic retransmission request is initiated by the GBS software. The Queue Manager periodically sends a list of files that have been recently broadcast. The receive software compares this list against files it has received and automatically requests retransmission of files that were never received. A manual transmission request is one that is initiated by a user. Typically, a manual request is made when a file was received, but deleted, or the file was discarded due to the dispose rule settings.

The automatic retransmission capability provides a means for assured delivery of data files for those receive workstations with network connectivity to the BMC. If some form of network connectivity exists all receive workstations, files would not need to be broadcast multiple times. All files not properly received by a site would automatically be requested to be retransmitted providing assured file delivery.

5.1.2 Data Streams

In addition to the ability to broadcast data files, the Phase I software has the ability to broadcast data streams. These data streams are typically provided via a hardwire connection to the GBS Gateway or Queue Manager. The *streamClient* process reads data from the input source (e.g., a serial port), which is defined in a command line specified configuration file. The data from the input stream is broken into UDP packets when utilizing an IP based transport or sent via a specified VP/VC when using the ATM transport.

When the streaming data is received at the GBS receive workstation, the *streamServer* process puts the data packets back together and write them out on the tty port specified in the process' command line specified configuration file.

Several variations of this process have been implemented so that the streaming data can be received from other applications and provided via standard input to the *streamClient* process. Additionally, variations have been made to streamServer so that when the data arrives at the GBS receive workstation, it can be piped to standard output so that it can feed to a local application instead a tty port.

Although the data streaming mechanisms in Phase I provide a basic streaming capability, they are very limited. There exists no way to monitor the configured streams at either the transmit or receive ends to determine if data is being passed. Additionally, a user interface needs to exist which would allow a GBS user to enable/disable data streams as well as provide the streaming data to multiple applications or workstations.

One of the major limitations of the existing data streaming software is the inability to accept data streams via the network instead of a hardwired connection. This ability would allow for the broadcast of data which would otherwise be difficult or impossible to broadcast. Examples of this usage would be for the broadcast of news wires and TLAM tracking data.

COTS software exists today which provide capabilities similar to what is needed here. It is very likely that this functionality can be completely replaced with COTS software.

5.1.3 Multimedia

GBS Phase I software has integrated COTS software to provide multimedia capabilities. This capability enables the receive suite to simultaneously display both classified and unclassified video on the receive workstation. This capability has been implemented using the Uniflix COTS product and the installation of a specialized video board in the Sun receive suite workstation.

Just as users can tune their IRD for displaying unclassified video on a television, a separate IRD has been included in the receive suite whose output is directed to the Sun workstation video board instead of television. This video is then displayable in a window on the Sun workstation.

The classified video is provided in a completely different manner. A workstation at the BMC has been configured to accept a classified video input. Through the use of the COTS software, this classified video is packaged into multicast IP packets and included in the GBS data channel broadcast. When these packets are received at the receive workstation, the COTS software reassembles the packets and displays the video in a window on the receive workstation.

The current implementation of classified video is limited to one classified video channel. The COTS software being used contains custom enhancements which defeats the purpose of using COTS software. With the increased use of video on LANS, the number of video software packages is increasing. The use of a COTS product to provide audio/video data transmission should be continued, however which COTS product is used should be re-evaluated. Also the mechanism used to display the video available on the workstation should be used limitations with the current implementation as well as the expected change in platforms.

5.1.4 Broadcast Status Messages

Data file traffic across the GBS broadcast does not maintain a sustained data rate because the data being broadcast is dependent on when the sources insert files into the broadcast. For this reason, a small but constant flow of status information is added to the broadcast so that the receive sites have feedback on the status of the broadcast at all times. This status information consists of a "heartbeat" plus textual status information which can be entered at the BMC to inform sites of upcoming events such as system outages.

This means of keeping sites informed of the broadcast status is simple but efficient. Many times when textual messages are entered at the BMC, they go unnoticed for some time. A more visible way of displaying these textual messages should be developed. One such means would be to displaying these messages in an alerting popup window.

5.1.5 Cryptographic Device Synchronization

When using the IP transport of the GBS broadcast, the cryptographic device at the receive workstation sometimes looses its synchronization. For this reason a process was developed in which data packets are transmitted which assists the device in regaining synchronization. This process is effective, but this is a case of software performing a job which is better handled by hardware. The advantage to a hardware solution is quicker response time. This problem should be reinvestigated for non-software solutions.

5.1.6 Exploitation Applications

GBS Phase I has been used to broadcast many different types of products, including imagery files (nitf), graphics files (gif/jpeg/tiff), and news articles (pdf). In order for the receive suite user to view these products, applications such as Matrix, XView and Acrobat Reader have been installed on the GBS receive suite. This was necessary because either the user did not have the necessary applications or they were

unable, for security accreditation reasons, to connect the GBS system to their local LAN where these exploitation applications exist.

It should not be the job of GBS to provide an application for viewing each type of file broadcast via GBS. GBS is a transport mechanism and not a data exploitation system. GBS should provide a means for getting files to the receive suite user so that users can then use other systems within their organization for accessing the files. There are many types of files available on the Internet, but we do not expect that our workstation will have applications for accessing each of these files. If this were the case, our workstation would have a dozens of different word processors, spreadsheets, video players, etc.

5.2 System Operation & Administration

5.2.1 "Lights Out" System

Both the BMC and receive suite portions of the GBS has been designed to operate with minimal operator intervention. At the BMC, operator intervention is only required for tasks such as backups and monitoring the queue status to make sure it is not overloaded. The latter event has occurred several times, and is usually the result of a run-away process repeatedly injecting the same files into the broadcast.

On the receive suite, a background process periodically runs to delete files which have "expired". Another background process ensures that the log files do not overrun the available disk space on the system.

The requirement to minimize periodic maintenance is one of the primary lesson learned from the Phase I system.

5.2.2 System Administration

The system administration of GBS is composed of two portions: UNIX system administration and the administration of GBS itself. UNIX system administration consists of tasks such as backups, account maintenance, peripheral configuration such as adding or deleting printers, and the administration of the file system. During Phase I, the administration of the file system has been the most demanding of these. This has been due to the limited disk space on the GBS machine and its use to maintain an archive of files broadcast. GBS version 2.1 has made attempts to resolve this problem with the addition of a process to perform the automatic deletion of data files once they have expired.

The administration of GBS itself consists primarily of maintaining rules for determining the disposition of received files and configuring access to received data files for other workstations such as a weather workstation wishing to processes all weather files received.

The system administration required for Phase I has proved to be too difficult for the users. This is one reason a strong preference exists for the use of Windows NT operating system for Phase II.

5.3 Miscellaneous System Capabilities

5.3.1 Executive/Alarms

The user interface was drastically enhanced for release 2.1. This enhancement included the integration of software to provide the receive suite user with an improved graphical interface for accessing the various applications comprising GBS.

Though this software has proved to be a satisfactory user interface, it should be replaced using an approach common across many different systems. The choices are to either make GBS software DII COE compliant or to utilize the approach provided with the operating system (e.g., Common Desktop Environment (CDE) for UNIX platforms). DII COE integration provides a set of common user tools available for both Solaris and Windows NT.

5.3.2 Automatic System Cleanup

The latest version of Phase I software includes an automatic system cleanup capability. This capability provides for the automatic truncation of GBS log files and for the deletion of "expired" data files. The automatic system cleanup process was added because of the disk on the receive suites were constantly filling up and the user did not necessary have the UNIX knowledge to correct the problem.

This process will hopefully become obsolete in Phase II, if the data products are delivered to the final destination instead of being cached on the GBS workstation.

5.3.3 Network Time Protocol

A shareware package which implements the Network Time Protocol (NTP) has been loosely integrated into the Phase I GBS software. The integration of this package provides a means for all GBS receive workstation to synchronize their time with a time being broadcast from the BMC. The source of the time being broadcast from the BMC is configurable, and is currently received via SIPRNET workstation with a Global Positioning Satellite (GPS) receiver.

The inclusion of this capability not only synchronizes the GBS receive workstation, but allows other workstation networked to the GBS receive workstation to be synchronized as well. This NTP software is available for a variety of hardware platforms and operating systems.

The inclusion of this software is a prime example of how GBS can leverage off COTS and GOTS software.

5.3.4 Over-the-Air Software Upgrade

Phase I software incorporates the capability of broadcasting files which can be automatically executed when received at GBS receive workstations. This capability has been used to upgrade receive suites when the situation has demanded it. This capability can also been used to perform system maintenance of one or more receive workstation which may not have network connectivity for accessing the machine via telnet.

In order to cause a file to be automatically executed when it is received, a specific option is specified when wrapping the file. The use of this option is closely held so as to control the access to this feature. Phase II software should provide a similar capability, but should put much tighter control of its use.